



# COMPUTER CENTRE BULLETIN



BULLETIN MAILING LIST

The Bulletin is distributed free of charge to organizations, departments, and individuals. The mailing list is maintained on an annual basis for all subscribers.

To receive volume 6 of the Bulletin, please complete the following form, and return it as soon as possible to:

The Senior Operator,  
Computer Centre,  
University of Queensland,  
St Lucia 4067.

APPLICATION FOR ORGANIZATIONAL OR INDIVIDUAL CIRCULATION

ORGANIZATION or  
NAME

POSITION  
(if applicable)

ADDRESS

NUMBER OF COPIES

DATE

SIGNATURE



### USER REPRESENTATION ON THE COMPUTER CENTRE EXECUTIVE COMMITTEE

In 1971, the composition of the Computer Centre was reviewed with the intention of ensuring that users of Computer Centre's services were adequately represented.

As a representative from each department or each faculty would have resulted in too large a body, faculties were grouped and a number of representatives elected from each group, the number being related to usage by the particular group. In addition to departmental representatives, the revised composition includes four ex officio members and four other members. Professor Prentice, in recognition of his services in establishing computing in the university, is a personal member.

In 1972, the membership was increased by a further member to represent the computing interests of Administration.

Users of Computer Centre services are invited to discuss problems with their representatives so that they may be brought to the attention of the Executive Committee. Mr N. C. Watson, the member nominated by the Director of Post and Telegraphs of Queensland, would be pleased to represent the interests of external users of the Centre.

#### Membership 1972 is as follows:

Deputy Vice-Chancellor (Fabric and Finance)	Mr J.E. Ritchie
President of Professorial Board	Prof. Dorothy Hill
Registrar	Dr S.A. Rayner
Director of Computer Centre	Prof. G.A. Rose (acting)
Senator nominated by the Senate	Lt Col A.S. Gehrman

Representatives of each of the following three groups of faculties, nominated by the deans of the relevant faculties:

Arts	Prof. R.C. Gates (Chairman)
Commerce and Economics	(Economics)
Education	
Law	Dr A.R. Pike (Psychology)
Music	

Architecture  
Engineering  
Science

Prof. D.J. Nicklin (Chem. Eng)  
Mr R.D. Nilsson (Civil Eng)  
Dr K.L. Jones (Physics)

Agriculture  
Dentistry  
Medicine  
Veterinary Science

Mr A.W. Beattie  
(Vet. Animal Husbandry)  
Dr H. Silverstone  
(Soc. & Prev. Medicine)

Head of the Department of Computer Science or his deputy  
Mr R.E. Kelly

One member of the staff of the Centre nominated by the Director  
Mr J.D. Noad

One member nominated by the director of Posts and Telegraphs for  
Queensland

Mr N.C. Watson

#### CHRISTMAS SHUT DOWN

The Computer Centre will shut down on 22 December 1972 and remain closed during the intervening days between Christmas and New Year. No further work will be processed after 11.00 p.m. on Thursday 21 December. However, the Centre will remain open until 1 p.m. on Friday 22 December to allow staff to finalize end-of-year accounting and users to collect any work or card punching. The Centre will reopen at 9 a.m. on Tuesday 2 January 1973.

#### POLICY FOR REVISION OF COMPUTER CENTRE SOFTWARE

One of the many functions of the staff of the Centre is the maintenance of supported software. This work covers two broad areas.

- (a) Checking of reported errors, and implementation of corrections either advised by the supplier or developed locally.

- (b) Reviewing, testing and implementing new versions of system programs that become available.

In determining a policy on the frequency of updating software, the following factors must be considered.

- (a) Before release as a supported package, each new version or correction level of a program requires extensive testing to ensure that, as far as possible, new features and corrections work as intended and that no new errors have been introduced. It is therefore impractical to correct each error as it is found, unless an error is such as to make the software completely inoperable.
- (b) Many clients have indicated that they would prefer stable software with known and documented errors, rather than software that is continually changing.
- (c) The manufacturer provides the Centre with software support (in terms of error correction) only on the latest version of a given item of software, and only for a limited period after that version becomes available. This provides some pressure to upgrade software frequently. However, in many cases, because of size or reliability problems, the Centre has been unable to adopt the latest versions.
- (d) The manufacturer has announced that some of the new software packages for the PDP-10 have been 'unbundled', i.e. the Centre will have to lease these items of software if they are to be made available. While at present this unbundling is fairly limited, future implementation of unbundled items must be financially justified.
- (e) Within the limited resources of the Centre, it is not possible to conduct detailed testing and analysis of every new software release from the supplier.

In an attempt to meet the diverse requirements given above the following policy on software maintenance has been adopted.

(a) Software Support

The Centre will provide support for only one version of any software or program package classified as type 1 or type 2 software (cf Bulletin vol. 4 no 5 p. 62) at any time.

(b) Software Error Correction

The documented errors and available corrections for each item of type 1 and 2 software will be reviewed at intervals of two months. Where there are either a large number of

outstanding corrections for the given item, or outstanding corrections of major significance to that item, these will be scheduled for implementation following the review.

(c) Adoption of New Versions of Software

New versions of type 1 and 2 software will be reviewed and considered for implementation every year. A new version should be implemented if it

- (i) provides significant new facilities
  - or (ii) gives greatly increased performance
  - or (iii) provides improved reliability
  - or (iv) is required by other changes
- and does not, as far as can be determined, introduce any new errors or problems.

PROPOSED CHANGE TO FORTRAN, COBOL, ALGOL & MACRO COMMANDS

[WN-109]

If a user wants a listing of his program, the default file used is the job output device, i.e. \$TTY for terminal users or \$LPT for batch.

Normally terminal users would prefer to have their output on a disk file so that they can TYPE or LIST it later.

The proposal is to change the default list file so that the command

F4(LIST) SOURCE

- (i) for a terminal user the file will go to disk, with the name  
SOURCE/LST
- (ii) for a batch user the file will go to \$LPT.

If you have any comments please write to the Centre as soon as possible.



COBOL AND ALGOL DATA FILES IN BATCH

[WN-107]

It is important to note that Cobol and Algol, when accepting data from the input device (i.e. cards) do not under all circumstances recognize the file separator card as terminating the file, and may attempt to read beyond it. This situation may cause serious problems with the student accounting system. It is recommended that data be copied to a disk file and input taken from that file.

In summary, do not put data for your Cobol or Algol programs after the RUN command but copy them to a disk file prior to execution of your program.

SYSTEM PROGRAMS

[WN-108]

New versions of system programs were released on Monday 25 September.

- (a) PIP version 30.57 replaced Pip version 30.55 on the system directory. The only observable difference is the removal of the '\$' which preceded library directory names.
- (b) Finish version 6.0 replaced version 5.6. Changes were made to reduce the size from 3K to 2K.

The student accounting system in batch is protected against '.EOJ' in a student job, and absence of the final '.QUIT' command.

The message

.EOJ IS ILLEGAL IN THIS CONTEXT,  
.QUIT IS ASSUMED

is given and a QUIT command is performed instead.

B5-11  
24Oct72

### DEC PLOTTER ROUTINES

[WN-109]

When the CalComp plotter routines were released it was stated that the DEC routines would be removed from the Fortran library. This was done on Monday 9 October. It is now obligatory when using the DEC plotting routines to include PLOT.DEC(LIB) in the Run or Load command.

### PLOT

[WN-107]

An error in the PLOT routine in the CalComp library file has been corrected. The error resulted in incorrect pen positioning when a large movement in the negative X direction was required.

### LINEAR

[WN-109]

Linear, a linear programming program, has been placed on the OR directory. At present, limited copies of a writeup are available from the Centre upon request but this material will shortly be generally available on microfiche. The writeup does contain an error; the command to initiate the program should be 'OR.LINEAR' and not just 'LINEAR'.

### SNOBOL 3

[WN-107]

The Snobol 3 Interpreter has been released on the LANG directory.

LIBRARY ACCESSIONS

SYDNEY UNIVERSITY, Department of Architectural Science  
Computer report, no 9. 1970 and onwards  
(721.08 SYD Arch.)

- Data processing in biology and geology.  
1971 (574.028 DAT Geol.)

AUSTRALIAN WATER RESOURCES COUNCIL  
Standards for interchange of surface  
hydrologic data on computer media. 1970  
(Qto029.7 AUS Engin.)

HENLEY, John Patrick Computer based library and information  
systems. 1970 (020.18 HEN Clin.)

COWAN, T.K. Management accounting, objectives  
systems analysis of relevant costs.  
1971 (HF5635.C8 U/G)

ROSE, Michael Computers, managers and society. 1969  
(HF5548.2.R625 U/G)

ROTHERY, Brian The myth of the computer. 1971  
(HF5548.2.R634 Main)

- Computers in the classroom. 1970  
(371.3944 COM Educ.)

JOHNSTON, F.J.J. Decision tables in data processing.  
1970 (Qto001.64 JOH Engin.)

SLAGLE, James R. Artificial intelligence. 1971 (001.535  
SLA Engin.)

CANADIAN INSTITUTE OF CHARTERED ACCOUNTANTS,  
Study Group on Computer Control and Audit Guidelines  
Computer control guidelines. 1970  
(QtoQA76.5.C35 Main)

LINDSEY, C.H. Informal introduction to Algol 68. 1971  
(651.8 LIN Engin.)

SILVER, Gerald A. Simplified Fortran IV programming. 1971  
(651.8 SIL Engin.)

STENBERG, Warren Calculus; a computer oriented  
presentation, Teacher's commentary.  
1970 (QA303.S842 Maths)

B5-11  
24Oct72

- international journal of bio-medical computing, vol. 1. 1970 and onwards (RA409.5.I5 Pharm.)
- GREAT BRITAIN CIVIL SERVICE DEPARTMENT  
Computers in central government - ten years ahead. 1971 (JN329.A8G7 Main)
- HYVARINEN, Lassi P. Information theory for systems engineers. 1970 (Q360.H9 Maths)
- BARRON, David William Computer operating systems. 1971 (621.38 1953 BAR Engin.)
- GASS, Saul I. Linear programming. 1969 (T57.74.G3 Maths)
- INTERNATIONAL IFIP/IFAC PROLAMAT CONFERENCE, 1st, Rome 1969  
Numerical control programming languages. 1970 (621.902 INT Engin.)
- SYMPOSIUM ON COMPUTER AND INFORMATION SCIENCES  
3rd, Miami Beach, Florida 1969  
Software engineering. 1970 (651.8 SYM Engin.)
- DITRI, Arnold E. Managing the EDP function. 1971 (HF5548.2.D54 Main)
- SWEENEY, Robert B. The use of computers in accounting. 1971 (QtoHF5679.S93 Main)
- MICHIGAN UNIVERSITY, Department of Geography  
Selected computer programs. 1970 (GA23.M53 Main)
- BARKER, P.J. Basic Computer Studies. 1970 (651.8 BAR Engin.)
- BOUDAREI, Rene J. Dynamic programming and its application to optimal control. 1971 (QA402.3.B67134 Maths)
- BROWN, Kirk W. Computer program for plotting time dependent data with instruction and examples. 1969 (Qto510.7834 BRO Maths)
- DAVIS, Gordon Bitter Elementary Cobol programming. 1971 (QtoQA76.5.D285 Main)

DAVIS, Gordon Bitter	Introduction to electronic computers. 1971 (QA76.5.D29 Main)
GREGORY, Robert Todd	A collection of matrices for testing computational algorithms. 1969 (QtoQA263.G68 Maths)
GROUSE, P.J.	An introduction to computer programming in PL/I. 1971 (QtoQA76.5.G7 Main)
SCIENCE RESEARCH COUNCIL	ATLAS SYMPOSIUM, 2nd, Oxford 1969 Computers in number theory. 1971 (QA241.S37 Maths)
TRAINING SYSTEMS INC.	Computer numbering systems and binary arithmetic. 1965 (511.1 TRA Engin.)
WELLS, Mark B.	Elements of combinatorial computing. 1971 (511.6 WEL Engin.)
WILKINS, B.R.	Analogue and iterative methods in computation, simulation and control. 1970 (621.381957 WIL Engin.)

#### COMPUTER CLUB PROGRAMMING SERVICE

This advertisement has been placed in the Bulletin at the request of the Computer Club as being of possible interest to the Centre's users. The proposed service is not connected with Computer Centre services.

The Computer Club feels that there are many people and departments within the University who wish to employ programmers on a short term basis to develop specific programs. At the moment however, no program development service is available. This seems an ideal opportunity for members of the club to gain good experience, earn a little money, and provide a useful service to people and departments in the University.

Naturally, we cannot provide professional programmers, and clients will realize that our members are students in the computer field. Nor can we make recommendations to clients of a person's programming ability. This does not mean that there are not good programmers within the club, just that the executive cannot maintain close enough contact with all club members to guarantee the standard of skill they may have reached.

B5-11  
24Oct72

Prospective clients are asked to submit to the club written specifications of the job concerned. If it is felt these are inadequate the club may request further clarification.

These 'advertisements' will receive the club stamp, and be placed on the club notice board for perusal by club members. At the moment the board is situated outside B18, Main Engineering Building. Any notices without the club stamp will be removed.

Matters of payment and conditions under which the job is computed are strictly between the client and the member. The club is unable to accept any responsibility at all.

It is hoped that this service will fulfil a real need existing at the moment. Prospective clients may submit jobs to the club at this address:

Programming Service  
Computer Club  
c/- U. Q. Students' Union  
St Lucia 4067

#### A FORTRAN FUNCTION USEFUL IN MATRIX OPERATIONS

W.J. Whiten

The following Fortran function has been found of considerable use in programming matrix operations and similar applications.

```
      FUNCTION SPD(N,A,NA,B,NB)
C      CALCULATE THE SUM OF THE PRODUCTS
C          A(1 + NA*I) * B(1 + NB*I)
C      FOR I=0,1,2, ..., N-1
C
C      REAL A(100),B(100)
C
C      SPD=0.
C      IF (N.LE.0) RETURN
C      IA=1
C      IB=1
```

-----

Mr Whiten is research officer at the Julius Kruttschnitt Mineral Research Centre.

```

C
  DO 10 I=1,N
    SPD=SPD + A(IA) * B(IB)
    IA=IA+NA
10  IB=IB+NB
    RETURN
  END

```

This subprogram will replace the inner loop of many matrix operations. The test for the sum of zero elements is useful when back-substitutions in triangular matrices are required.

example:

```

C      DIMENSION A(10,20),B(20,12),C(10,12),X(10),D(10)
C      THE ACTUAL SIZE OF ARRAYS BEING USED IN THIS RUN
C      IS A(NI,NK),B(NK,NJ),C(NI,NJ),X(NI),D(NJ)
C
C      . . .
C      MATRIX PRODUCT C=A*B
C      DO 10 I=1,NI
C      DO 10 J=1,NJ
10    C(I,J)=SPD(NK,A(I,1),20,B(1,J),1)
C
C      . . .
C      SOLUTION OF EQUATION C*X=D
C      WHERE C IS LOWER TRIANGULAR
C      DO 100 I=1,NI
20    X(I)=(D(I) - SPD(I-1,C(I,1),12,X(I),1))/C(I,I)

```

In this example statement no 10 replaces four lines of code and statement 20 replaces five lines of code.

When using this subroutine for multidimensional arrays it is necessary to remember the manner in which these are held in storage (first subscript varies most rapidly i.e. column-wise for matrices). Then the actual elements of the first product and distances (which are 1 or derived from the DIMENSION statement) between the required elements are used in the calling sequence of SPD.

The accuracy of this function, and usually the operations using it, may be improved by forming the sum of products in double precision and then converting the result to single precision before the return statement. Analysis of rounding errors in poorly conditioned problems has shown that this use of double precision has much to recommend it.

B5-11  
24Oct72

As in many programs this function becomes the place where most of the arithmetical operations are performed. On some machines a decrease in total run time can be obtained by using machine language for this function.

#### USER PROGRAMS

In recent issues of the Bulletin there have been descriptions of a number of programs belonging to users who have kindly made their programs available for general use. The following list is a short summary of the programs that have been so released.

<u>program</u>	<u>function</u>	<u>description</u>	<u>contact</u>
94.LPLOT/REL	line printer plotting program	B4-12 6Dec71	Mr W. Whiten J.K.M.R.C.*
94.TRANS/REL	multiple spline regression program	B4-12 6Dec71	Mr W. Whiten J.K.M.R.C.
94.CURFIT/REL	program for curve fitting and data adjustment	B5-5 2May72	Mr M. J. Lees J.K.M.R.C.
525.FRED/REL	free read for PDP-10 Fortran	B5-7 5Jul72	Mr A.J. Mortimer Elect. Eng.
177.EVAL/SAV	multichoice test marking scheme	B5-10 24Sep72	Mr D. Woodrow St Peter's College Brisbane
6.CORREL/SAV	correlation program	B5-11 24Oct72	Mr W. Gout Social and Prev. Med.

\* Julius Kruttschitt Mineral Research Centre



B5-11  
24Oct72

525.JOBERR/SHR elementary batch job conditional testing and control program

B5-11  
24Oct72

Mr A.J. Mortimer  
Elect. Eng

MATRIX.SMIS Symbolic Matrix Interpretive Scheme

B4-8 & 9  
6Sep71

Mr R.D. Nilsson  
Civil Eng

166.DUMP/SHR file examining and updating program

258.DUMP/DOC  
(warning -  
app. 19K wds)

Mr C.G. Wilson  
278 Montagu Road  
West End

### CORRELATION PROGRAM

W. Gout

#### 1 GENERAL DESCRIPTION

This program produces means and standard deviations and the correlation matrix for up to 50 variables. The following formulae are used:

$$(i) \quad \bar{x}_i = \sum_{j=1}^n x_{ij}$$

$$(ii) \quad SD_i = \sqrt{\frac{\sum_{j=1}^n (x_{ij} - \bar{x}_i)^2}{n - 1}}$$

$$(iii) \quad r_{ij} = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{(n - 1) \cdot SD_i \cdot SD_j}$$

where  $\bar{x}_i$  is the mean for the  $i$ th variable

$x_{ij}$  is the  $j$ th observation for the  $i$ th variable

---

Mr Gout is a programmer in the Department of Social and Preventive Medicine.

$n$  is the sample size

$SD_i$  is the standard deviation for the  $i$ th variable

$r_{ij}$  is the correlation coefficient for the  $i$ th and  $j$ th variables.

## 2 INPUT

The raw data, with the restriction that every record must contain an observation for each variable. Short (10 characters) descriptions of each variable are also entered.

## 3 OUTPUT

- (a) table of means and standard deviations for each variable
- (b) correlation matrix
- (c) (optional) if desired, the above-diagonal elements of the correlation matrix may be output onto a file for subsequent processing by another program. The format used is (10F8.5).

## 4 OPERATING INSTRUCTIONS

### 4.1 Teletype Use

Type 6.CORREL and await further instructions from the program. When requested, enter:

- (a) the input filename (blank if input is from teletype)
- (b) the output filename (blank if output is to teletype)
- (c) the filename for the labels, (if labels are to be entered from the teletype, type RETURN)
- (d) the name of the file for output of the above diagonal elements of the correlation matrix as in 3(c) above. If this is not required, type RETURN.
- (e) a 70 character title

- (f) the number of variables
- (g) the data format. This must be F-type format, and may contain up to 70 characters.
- (h) the labels for the variables (unless they are on a file), each label may be up to 10 characters in length. Enter one per line.
- (i) the data (unless data are on a file) according to the format specified. After all the data have been entered, type tZ (depress control, and then, keeping the control key depressed, type Z).

After the results have been produced, further sets of data may be processed by typing YES when ANY MORE DATA? is typed, then the input filename (as in (a) above), and then proceeding from (c) above.

#### 4.2 Batch Use

The following deck setup is used (punchings start at column 1):

- card 1 : .6.CORREL
- card 2 : input filename (blank if data on cards).
- card 3 : output filename (blank if output to line printer)
- card 4 : filename for labels (blank if cards are used)
- card 5 : output filename for output of above-diagonal elements of correlation matrix. If not required this card should be blank.
- card 6 : title card, the first 70 columns may be used.
- card 7 : number of variables
- card 8 : data format, the format must be F-type, and the first 70 columns may be used.
- card 9 : labels (if labels are on cards)
- etc. one card per variable, columns 1-10 may be used
- data cards } if data are on cards
- file separator card }

last card : blank (except if another run follows, in which case a card with YES is used, followed by a card with the input filename (as in card 2), then repeat from card 4 above).

## 5 GENERAL

A correlation coefficient of 9.999 is printed if division by zero occurs (e.g. if the variance of one of the variables is zero).

If some observations are missing, it may be possible to use the program CORZER.

## 6 RESTRICTIONS

The number of variables must be between 2 and 50 inclusive.

## 7 COST

Depends on method of operation (batch is cheaper). As an example, through batch: 7 variables with 60 data cards would cost about 20cts; 47 variables with 20 observations per variable cost about \$1.75.

## 8 TECHNICAL DATA

CORREL is a Fortran IV program using 5K core (plus 4K for the Fortran operating system).

### JOBERR

A.J. Mortimer

## 1 INTRODUCTION

The program JOBERR has been written to provide an elementary form

-----

Mr Mortimer is a post graduate student with the Department of Electrical Engineering.

of conditional testing and job control for batch processing. This document contains instructions for using this program to control the running of batch jobs.

## 2 DESCRIPTION

A brief description of the process used on the PDP-10 to detect compilation and other program errors is in order. In the user's core area, the first 140(octal) locations are reserved for storing information pertaining to the user's job (such as starting address, program version number, core limits, and other similar information). One location (.JBERR) in this area is used to keep a count of errors detected in the user's program. In particular, the right half of this location is set to a nonzero value by the system compilers and assemblers on detection of compilation and assembly errors. The value of .JBERR is then used by programs such as the loader to decide whether to execute the user's program or not.

Unfortunately, this procedure provides no facilities to allow the user to control the running of his program, and furthermore, results in loss of money since the loader actually loads the program before deciding whether or not to continue execution.

The program JOBERR provides an elementary but effective method of testing the status of .JBERR and executing some action conditional on that status, thus allowing the user to decide for himself what action is desirable where compilation and other such errors are detected.

In the following description, .JBERR is considered to be set if its right half is nonzero, and reset if its right half is zero.

## 3 METHOD OF USE

### 3.1 CALLING SEQUENCE

The program is held on project number 525 and may be called from that area by the command:

525.JOBERR <commands>

Note that any commands to the program must appear in the one record. There are no facilities available for extension to a second record.

### 3.2 COMMAND FORMAT

The <commands> have the following form:

<commands> = <command1>b<command2>b ... b<commandi>

where b stands for one or more blanks, tabs, or commas.

The command list is terminated by any break character, by the end of the record, or by a semi-colon. Comments may therefore appear in the record, following a semi-colon.

The individual commands may have any of the following forms:

	SET
	RESET
	CLEAR
<commandi> =	NOCLEAR
	SKIP <n>
	GOTO <label>
	IFSET (IFS)
	IFNSET (IFN)

The command forms shown in parentheses are permitted abbreviations. The command actions are described below.

### 3.3 COMMANDS

#### (a) controlling the value of .JBERR

- (i) SET  
sets the right half of .JBERR to one.
- (ii) RESET  
clears the right half of .JBERR

#### (b) controlling the loadlist

- (i) CLEAR  
clears the current loadlist.
- (ii) NOCLEAR  
used to complement the clear command, leaves the

loadlist as is.

(c) branch instructions

- (i) SKIP <n>  
tells the program to swallow <n> following records. If <n> is not given, it is assumed to be 1. <n>=0 may be given to complement an earlier SKIP command if so desired.
- (ii) GOTO <label>  
tells the program to swallow records, until it finds one starting with a semi-colon in column one, and with the <label> immediately following. The <label> may consist of 1 to 6 alphanumeric characters (A-Z, 0-9) and must start with an alphabetic character (A-Z). If no label is given, any previous label is deleted.

Note that one or more blanks, tabs, or commas must appear between the command and its argument. It is also permissible for conditional commands to appear between the command and the argument.

For both the SKIP and the GOTO command, no more than 3000 records may be skipped, and skipping is always terminated by a .QUIT or an .IDENT record. If the skipping is terminated in this manner, the program forces a rescan of the record, thus allowing Batch to handle the next student job correctly.

(d) conditional instructions

- (i) IFSET or IFS  
tests .JBERR and succeeds if it was set, but fails if it was not set.
- (ii) IFNSET or IFN  
tests .JBERR and succeeds if it was not set, but fails if it was set.

3.4 SCANNING RULES

- (a) The command string is scanned and interpreted by the program, scanning from left to right. If a conditional command (IFS, IFN) is found, the relevant test is carried out immediately. All other commands are stored for later execution.

- (b) The command scan is terminated by any error detected during the scan, or by any break character (lf, vt, ff, altmode, ↑G, ↑Z).
- (c) If a conditional command fails, the command scan continues, ignoring all command strings, until either:
  - (i) the scan terminates as per (b) above or
  - (ii) a dollar sign is found. On finding the dollar (\$), normal scanning is resumed. The dollar symbol is treated as though it was a blank in all other cases.

### 3.5 PROGRAM ACTION

On termination of the scan, the latest command of each type is executed. Hence, the latest command in each of the groups considered above is used. Thus it can be seen that the program provides a reasonably effective form of conditional control for batch processing.

## 4 EXAMPLES

### (a) simple example

```
<cards>
.525.JOBERR IFN SKIP                ;here if error - kills job
.eoj                                ;here if no error - job continues as normal
<cards>
```

### (b) example using goto and both conditionals

```
<cards>
.525.JOBERR GOTO IFN TST1 $IFS TST2 ;note that the ifn
                                   ;could be omitted
<cards>                            ;these cards always ignored
;tst1                             here if .JBERR not set
<cards>
;tst2                             here if .JBERR set
<cards>
```

### (c) more complicated skip instruction

```
<cards>
.525.JOBERR IFS SKIP 20 CLEAR RESET
;here if .JBERR not set
```



B5-11  
24Oct72

<19 cards>  
;here if set  
<etc...>

